

基于 PCTSP 的热轧单元计划模型与算法

刘士新¹, 周山长¹, 宋健海², 王梦光¹

(1. 东北大学 信息科学与工程学院, 辽宁 沈阳 110004; 2. 上海宝信软件股份有限公司 MES 事业部, 上海 201900)

摘要: 根据钢铁企业热轧产品生产工艺约束条件, 将热轧生产轧制单元计划模型归结为奖金收集旅行商问题, 设计了蚁群最优化算法对模型进行求解. 引用某钢铁企业热轧生产轧制单元计划编制的实际问题对模型和算法进行了验证, 并与遗传算法的求解结果进行了对比. 实验结果表明模型和算法的优化效果和时间效率都是令人满意的. 该模型和算法经过改进后可应用到包含多个轧制单元计划的轧制批量计划优化问题中.

关键词: 轧制单元计划; 奖金收集旅行商问题; 蚁群最优化; 遗传算法

中图分类号: C934

文献标识码: A

Prize collecting traveling salesman problem based model and algorithm for hot strip rolling unit planning

LIU Shi-xin¹, ZHOU Shan-chang¹, SONG Jian-hai², WANG Meng-guang¹

(1. School of Information Science & Engineering, Northeastern University, Shenyang Liaoning 110004, China;

2. MES Business Department, Shanghai Baosight Software Limited Company, Shanghai 201900, China)

Abstract: According to the process programs and constraints of hot rolling strip production in a steel plant, a prize collecting traveling salesman problem (PCTSP) based model for hot strip rolling unit planning is presented, an ant colony optimization (ACO) algorithm is designed to solve it. A practical instance in a steel plant is cited for testing the effectiveness of the model and algorithm. The plan obtained by the presented algorithm is compared with that obtained by a tested genetic algorithm. Experimental results show that the effectiveness and efficiency of the model and algorithm are satisfactory. With appropriate modifications, the model and algorithm can be applied to hot strip rolling lot planning problem which includes multiple rolling units inside.

Key words: hot strip rolling unit planning; PCTSP (prize collecting traveling salesman problem); ACO (ant colony optimization); GA (genetic algorithm)

1 引言 (Introduction)

热轧带钢是钢铁产品的重要组成部分. 热轧带钢生产管理的一个主要任务是编制合理的轧制单元计划, 并由多个轧制单元计划组成轧制批量计划. 由于生产过程的高度复杂性, 轧制计划的好坏直接影响产品质量、轧辊磨损成本和生产效率. 因此, 轧制单元计划优化问题一直吸引着众多业者和研究者的研究和关注^[1~3]. 本文综合考虑了热轧带钢生产过程中的工艺规程及约束, 针对轧制单元计划优化问题建立了基于奖金收集旅行商问题 (Prize Collecting Traveling Salesman Problem, PCTSP)^[4] 的热轧轧制单元计划优化模型, 设计了蚁群最优化 (Ant Colony Optimization, ACO) 算法^[5,6] 对模型进行求解.

2 数学模型 (Mathematical model)

2.1 工艺约束 (Process program constraints)

一个完整的轧制单元由烫辊材和主体材两部分组成, 单元中的板坯编排在宽度上呈“双梯形结构”. 烫辊材主要用于加热轧辊, 由于所需板坯数量少, 因此不在本文优化范围之内. 主体材是轧制单元的主要组成部分, 编制计划时通常要满足以下规程: 1) 板坯在硬度和厚度上要求平稳变化; 2) 带钢宽度总体上呈递减变化 (允许局部递增变化), 并且具有相同宽度的带钢连续轧制长度不能超过某一限量; 3) 整个轧制单元总轧制长度应小于轧辊的最大轧制长度.

轧制单元计划编制的过程可以看做是根据轧制规程约束,对候选板坯进行选择 and 排序的过程. 为了减少换辊次数,降低生产成本、提高生产效率,要求尽可能选择更多的板坯编入一个轧制单元. 为了降低轧辊磨损、保证产品质量,需要根据宽度、厚度、硬度等属性对板坯进行优化排序,最小化板坯在宽度、厚度、硬度上的跳跃对轧辊造成的磨损,并对选入的板坯在同宽轧制长度和总轧制长度上加以限制. 由以上分析发现,轧制单元优化过程类似 PCTSP 求解过程,因此,本文基于 PCTSP 建立了轧制单元优化模型.

2.2 优化模型 (Optimizing model)

将板坯看作 PCTSP 中的城市,引入如下符号,可基于 PCTSP 建立轧制单元计划优化模型. 记 V 为全部板坯集合, $i \in V$ 对应板坯 i , $n = |V|$ 为板坯总数, $i = 0$ 对应虚拟板坯 0. $V_c \subseteq V$ 为被编入轧制单元计划的板坯集合. c_{ij}^t, c_{ij}^h 和 c_{ij}^w 分别为板坯 i 到板坯 j 的厚度、硬度和宽度跳跃惩罚, 参数设置参见文献 [1], 则 $c_{ij} = c_{ij}^t + c_{ij}^h + c_{ij}^w$ 为板坯 i 到板坯 j 的总惩罚值, 对应由城市 i 到城市 j 的旅行费用, 其中 $c_{0i} = c_{0i} = 0, i \in V \setminus \{0\}$. p_i 为板坯 i 的轧制长度, 对应板坯 i 被编入计划时获得的奖金, $p_0 = 0$. γ_i 为板坯 i 未被编入轧制计划时受到的惩罚, $\gamma_0 = +\infty$. $B_{\min}$ 和 $B^{\max}$ 分别为奖金下限和上限, 对应一个轧制单元的最小轧制长度和最大轧制长度. $K^s = \{i_1, \dots, i_{|K^s|}\}, s = 1, \dots, Q$, 为编入计划、同属宽度系列 s 且连续轧制的最长板坯序列. $R^s, s = 1, \dots, Q$, 为属于宽度系列 s 的板坯的最大连续轧制长度. $x_{ij} = 1$, 如果 $i, j \in V_c$ 且在计划中板坯 j 紧接在板坯 i 后轧制, 否则 $x_{ij} = 0$. $y_i = 1$, 如果 $i \in V_c$, 否则 $y_i = 0$. 则热轧轧制单元计划优化模型描述如下:

$$\min Z = \sum_{i \in V} \sum_{j \in V} c_{ij} \cdot x_{ij} + \sum_{i \in V} \gamma_i \cdot (1 - y_i), \quad (1)$$

$$\text{s. t. } \sum_{j \in V} x_{ij} = y_i, \quad \forall i \in V, \quad (2)$$

$$\sum_{i \in V} x_{ij} = y_j, \quad \forall j \in V, \quad (3)$$

$$y_0 = 1, \quad (4)$$

$$\sum_{i \in V} p_i \cdot y_i \geq B_{\min}, \quad (5)$$

$$\sum_{i \in V} p_i \cdot y_i \leq B^{\max}, \quad (6)$$

$$\sum_{k=1}^{|K^s|} p_{i_k} \leq R^s, \quad s = 1, \dots, Q, \quad (7)$$

$$\sum_{i \in S} \sum_{j \in V \setminus S} x_{ij} \geq y_h,$$

$$\forall h \in V \setminus 0 \text{ 和 } \forall S \subset V | 0 \in S, h \in V \setminus S. \quad (8)$$

其中目标函数(1)代表板坯厚度、硬度和宽度跳跃惩罚以及因未编入计划板坯而受到的惩罚值之和最小; 约束(2)表示轧制计划中, 板坯 i 后有且只有一块板坯; 约束(3)表示轧制计划中, 板坯 j 前有且只有一块板坯; 约束(4)表示虚拟板坯 0 一定被编入计划; 约束(5)表示轧制单元最小轧制长度限制; 约束(6)表示轧制单元最大轧制长度限制; 约束(7)表示同宽轧制长度限制; 约束(8)保证任何编入计划的板坯都和板坯 0 相连, 且避免了子回路发生. 在实际应用中, 模型(1) ~ (8) 属于大规模组合优化问题, 求解困难. 本文针对模型特点, 设计了 ACO 求解算法.

3 优化算法 (Optimizing algorithm)

ACO 是近年来出现的一种元启发式算法, 在组合优化问题中获得了广泛的成功应用^[5-7]. 在本文 ACO 算法中, 板坯作为蚂蚁要访问的节点, 板坯之间的转移作为节点之间的弧, 板坯 i 到板坯 j 的厚度、硬度和宽度跳跃总惩罚值 c_{ij} 对应由节点 i 到节点 j 的转移费用. 如果相邻两块板坯因具有相同的厚度、硬度和宽度而转移费用为 0, 则将转移费用设置为较小值 0.01. 求解过程中每个蚂蚁按照如下方法构造旅行路径: 从虚拟板坯 0 出发, 优先选择具有较大宽度的板坯作为主体材的第 1 块板坯, 然后根据弧上信息素浓度 τ_{ij} 、启发优先系数 η_{ij} 以及约束条件随机选择下一节点, 在获得的奖金达到 $B_{\min}$ 后依概率 P 返回起始节点 0, 当获得奖金达到 $B^{\max}$ 后, 则必须返回起始节点 0.

以上求解过程保证了模型中约束(2) ~ (6)(8) 得到满足, 但无法满足约束(7). 采用惩罚函数法将约束(7)转换到目标函数中, 形成如下形式的目标函数:

$$\min Z' = \sum_{i \in V} \sum_{j \in V} c_{ij} \cdot x_{ij} + \sum_{i \in V} \gamma_i \cdot (1 - y_i) + \lambda \cdot \sum_{s=1}^Q \max \{0, (\sum_{k=1}^{|K^s|} p_{i_k} - R^s)\}. \quad (9)$$

其中 λ 为约束(7) 不满足惩罚系数.

引入以下符号: w_i 为板坯 i 的宽度. $w^{\max}$ 为所有候选板坯的最大宽度. m 为系统中蚂蚁的数量. k 为算法中蚂蚁的编号. kl 为蚂蚁 k 当前所处的节点. τ_{ij} 为某一求解状态下弧 (i, j) 上的信息素浓度. η_{ij} 为某一求解状态下蚂蚁由节点 i 转移到 j 的启发优先系数, 当蚂蚁处于节点 0 时, $\eta_{0i} = w_i / w^{\max}$, 当蚂蚁处于节点 i 时, $\eta_{ij} = 1 / c_{ij}$. $M_k(i)$ 为处于节点 i 的蚂蚁 k

下一步可以访问的节点集合, 根据问题约束确定. α 为信息素浓度 τ_{ij} 的相对重要性度量. β 为启发优先系数 η_{ij} 的相对重要性度量. ρ 为信息素挥发系数. k_{gb} 为从算法开始执行到目前为止产生了最好解的蚂蚁编号. 则在算法中处于节点 0 的蚂蚁 k 依概率 $p_k(0, i)$ 选择下一节点 i

$$p_k(0, i) = \begin{cases} \frac{[\tau_{0i}]^\alpha \cdot [\eta_{0i}]^\beta}{\sum_{z \in M_k(0)} [\tau_{0z}]^\alpha \cdot [\eta_{0z}]^\beta}, & j \in M_k(i), \\ 0, & \text{其他.} \end{cases} \quad (10)$$

处于节点 $i, i \neq 0$ 的蚂蚁 k 依概率 $p_k(i, j)$ 选择下一节点 j

$$p_k(i, j) = \begin{cases} \frac{[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{z \in M_k(i)} [\tau_{iz}]^\alpha \cdot [\eta_{iz}]^\beta}, & j \in M_k(i), \\ 0, & \text{其他.} \end{cases} \quad (11)$$

在算法迭代过程中 τ_{ij} 的值依据式(12) 进行延迟更新

$$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij} + \Delta\tau_{ij}. \quad (12)$$

其中 $\Delta\tau_{ij}$ 为信息素更新量, 在每次迭代完成后按式(13) 计算

$$\Delta\tau_{ij} = \begin{cases} \frac{W}{f_{gb}}, & \text{如果}(i, j) \text{ 包含在蚂蚁 } k_{gb} \text{ 所经过的路径中,} \\ 0, & \text{其他.} \end{cases} \quad (13)$$

式中 W 为常数, f_{gb} 为蚂蚁 k_{gb} 产生的解的目标函数值, 根据式(9) 计算.

算法流程如过程 1 所示, 步骤 4 的终止条件为完成 MaxIter 次循环.

过程 1:

1° 初始化: $\tau_{ij} := \tau_0, i, j = 0, 1, \dots, n, i \neq j, T_{gb} := \Phi; \{T_{gb} \text{ 为蚂蚁 } k_{gb} \text{ 经过的节点集合}\}$

2° 设置每个蚂蚁的出发节点为 0, $T_k := \{0\}; \{T_k \text{ 为第 } k \text{ 个蚂蚁经过的节点集合}\}$

For $k := 1$ To m Do

依式(10) 为蚂蚁 k 选择第 1 块板坯;

End-For

For $k := 1$ To m Do

If $M_k(kl) = \Phi$ Then $T_k := T_k \cup \{0\}, M_k(0) = \Phi; \{ \text{返回节点 0, 完成解构造过程} \}$

If $\sum_{i \in T_k} p_k \geq B^{\max}$ Then $T_k := T_k \cup \{0\}, M_k(0) = \Phi;$

If $\sum_{i \in T_k} p_k \geq B_{\min}$ Then 依概率 P 返回节点 0, 即

$T_k := T_k \cup \{0\}, M_k(0) = \Phi;$

根据式(11) 选择下一节点 $kl;$

更新 $M_k(kl);$

$T_k := T_k \cup \{kl\}; \{T_k \text{ 元素按加入顺序排列}\}$

End-for

3° For $k := 1$ To m Do

依式(9) 计算每个蚂蚁构造的解的目标函数值;

End-for

如果当前最优解得到改进, 则更新 $T_{gb};$

针对每一弧 (i, j) , 依据式(12) 更新信息素浓度;

4° If 满足终止条件 Then 输出 $T_{gb};$

Else 转步 2.

4 实验结果 (Experimental results)

为了测试模型和算法的优化效果及时间效率, 以某钢铁企业 71 个实际生产合同 (板坯) 为原始数据, 应用本文的模型和算法编制热轧轧制单元计划. 算法应用 Java 语言编写, 运行于使用 Windows 2000 操作系统的 Pentium IV (2.6G/512M) PC 机上.

4.1 对照算法 (Comparative algorithm)

实验中, 针对本文模型设计了简单遗传算法^[8] 来求解同一问题作为对比. 遗传算法采用双链表编码方式, 个体 $I = (\pi, l)$ 由二进制链表 $\pi = (e_1, e_2, \dots, e_n)$ 和非重复自然数链表 $l = (j_1, j_2, \dots, j_n)$ 构成. 链表 π 中第 i 个基因 $e_i = 1$ 代表板坯 i 被编入轧制单元计划, $e_i = 0$ 代表板坯 i 不被编入轧制单元计划. 链表 l 中的基因值为板坯编号, 基因位置对应板坯在轧制单元计划中排列的先后顺序. 解码过程是依据板坯在链表 l 中的排列顺序选择在链表 π 中对应的基因值为 1 的板坯形成轧制单元计划. 如果某个个体通过以上解码过程得到的轧制单元计划满足约束(5) ~ (7), 称其为可行个体, 否则称其为不可行个体. 定义不可行个体的适值为 0, 可行个体的适值根据下式计算

$$f(I) = f^{\max} - Z(I), \quad (14)$$

其中 $f^{\max}$ 为当前一代种群中所有可行个体解码后得到的轧制单元计划依据式(1) 计算的目标函数值的最大值, $Z(I)$ 为由可行个体 I 解码后得到的轧制单元计划依据式(1) 计算的目标函数值.

遗传算法中, 交叉和变异算子针对链表 π 和 l 的特点分别设计. 对于链表 π 采用遗传算法中二进制编码典型的一点交叉和变异算子. 对于链表 l , 采用求解 TSP 问题时广泛应用的部分匹配交叉 (PMX) 算子和 2-opt 变异算子. 选择算子采用轮盘赌方式进行.

4.2 计算结果 (Computational results)

模型参数设置如下: $B_{\min} = 30$, $B^{\max} = 60$, $\gamma_i = 150 \times p_i$, $R^s = 10$, $s = 1, \dots, Q$. ACO 算法参数设置如下: $m = \{6, 10, 12, 20\}$, $\text{MaxIter} = 100$, $\alpha = 1$, $\beta = 2$, $\rho = 0.2$, $\tau_0 = 0.2$, $P = 0.002$. 遗传算法参数设置如下: 种群规模为 20, 迭代代数 100, 交叉概率为 0.95, 变异概率为 0.05.

实验中, ACO 算法在 m 的每个取值下运行 10 次, 遗传算法在相同参数设置下运行 10 次. 实验结果统计如表 1 所示. 表 1 中第 2 列为不同 m 值下 ACO 算法和遗传算法 10 次运行结果的目标函数值的平均值; 第 3 列是不同 m 值下 ACO 算法和遗传算法 10 次运行结果的目标函数值最小值; 第 4 列是不同 m 值下 ACO 算法和遗传算法每次平均运行 CPU 时间.

表 1 试验结果统计

Table 1 The statistical results of computational experiments

m	平均值	最小值	CPU/s
6	9795	9469	2.43
10	9435	9408	3.82
12	9420	9371	4.61
20	9405	9365	6.70
GA	9817	9758	7.82

由表 1 可见, 当人工蚂蚁数较少时, 比如 $m = 6$ 时, 虽然算法运行速度比较快, 但求解效果并不令人满意. 当人工蚂蚁数为 $m = 10$ 或 12 时, 算法所得的平均值和最小值相差较小, 说明算法运行性能稳定, 算法求解效果明显优于 $m = 6$ 时的求解效果. 当人工蚂蚁数 $m = 20$ 时, 算法运行性能稳定而且获得了最优解, 运行结果比人工蚂蚁数为 10 和 12 时的优化效果稍有改进, 但运行时间却分别增加了 75.39% 和 45.34%.

两种算法比较, ACO 算法在求解效果和时间效率方面均优于遗传算法. 原因在于: 在求解效果方面, ACO 算法在蚂蚁构造解的过程中很好地处理了约束(6)(8), 并应用惩罚函数法处理约束(7). 相比之下, 遗传算法处理约束(6)~(8)时显得过于刚性. 在运行效率方面, 虽然两种算法每次运行产生的解数量相同, 但由于遗传算法每次迭代要进行选择、交叉和变异等操作, 并且在适值计算时需要判断解的可行性. 因此, 遗传算法较 ACO 算法具有较高的时间复杂性.

5 结论 (Conclusions)

本文结合热轧带钢生产实际和热轧生产工艺约

束, 权衡热轧生产率、轧辊利用率和轧辊成本之间的关系, 建立了基于 PCTSP 的热轧轧制计划优化模型, 设计了求解模型的 ACO 算法. 实验结果表明本文的模型和算法是令人满意的. 实验中发现, 由于系统中蚂蚁数量的取值不同, 算法的运行效果和收敛性能将受到不同程度的影响. 本文模型和算法经过改进, 可以应用到包含多个轧制单元计划的轧制批量计划优化问题中.

参考文献 (References):

- [1] KOSIBA E D, WRIGHT J R. Discrete event sequence as a traveling salesman problem [J]. *Computers in Industry*, 1992, 19(2): 317-327.
- [2] TANG Lixin, LIU Jiyin, RONG Aiyang, et al. A multiple traveling salesman problem model for hot rolling scheduling in Shanghai Baoshan Iron & Steel Complex [J]. *European Journal of Operational Research*, 2000, 124(2): 267-282.
- [3] LOPEZ L, CARTER M W, GENDREAU M. The hot strip mill production scheduling problem: A tabu search approach [J]. *European Journal of Operational Research*, 1998, 106(2): 317-335.
- [4] BALAS E. The prize collecting traveling salesman problem [J]. *Networks*, 1989, 19(6): 621-636.
- [5] 刘士新, 宋健海, 唐加福. 蚁群最优化-模型、算法及应用综述 [J]. *系统工程学报*, 2004, 19(5): 45-51.
(LIU Shixin, SONG Jianhai, TANG Jiafu. Ant colony optimization review: Modelling, algorithms and applications [J]. *Journal of Systems Engineering*, 2004, 19(5): 45-51.)
- [6] DORIGO M, STUTZLE T. The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances [M]// *Handbook of Metaheuristics: International Series in Operations Research and Management Science*. Glover F, Kochenberger G Eds. Kluwer; Netherlands, Kluwer Academic Publishers, 2003.
- [7] 马良, 项培军. 蚂蚁算法在组合优化中的应用 [J]. *管理科学学报*, 2001, 4(2): 32-37.
(MA Liang, XIANG Peijun. Applications of the ant algorithm to combinatorial optimization [J]. *J of Management Sciences in China*, 2001, 4(2): 32-37.)
- [8] GOLDBERG D E. *Genetic Algorithms in Search, Optimization, and Machine Learning* [M]. Massachusetts: Addison-Wesley, 1989.

作者简介:

刘士新 (1968—), 男, 博士, 副教授, 从事制造执行系统、项目管理、最优化方法等研究, E-mail: sxliu@mail.neu.edu.cn;

周山长 (1976—), 男, 硕士研究生, 从事制造执行系统、最优化方法等研究, E-mail: sc_zhou2002@163.com;

宋健海 (1965—), 男, 博士, 上海宝信软件股份有限公司高级工程师, 从事钢铁企业 ERP、MES 及企业信息系统整体解决方案等研究, E-mail: songjianhai@baosight.com;

王梦光 (1936—), 女, 教授, 博士生导师, 从事生产计划与调度、智能优化算法等研究, E-mail: mgwang@mail.neu.edu.cn.